

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
18 October 2001 (18.10.2001)

PCT

(10) International Publication Number  
**WO 01/78065 A1**

- (51) International Patent Classification<sup>7</sup>: **G10L 15/22**, 15/26
- (21) International Application Number: PCT/US01/11138
- (22) International Filing Date: 6 April 2001 (06.04.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
09/543,790 6 April 2000 (06.04.2000) US
- (63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:  
US 09/543,790 (CON)  
Filed on 6 April 2000 (06.04.2000)
- (74) Agents: SZE, James, Y. et al.; Pillsbury Winthrop LLP, P.O. Box 7880, 50 Fremont Street, San Francisco, CA 94120-7880 (US).
- (81) Designated States (*national*): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).
- Published:**  
— with international search report  
— before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments
- (71) Applicant (*for all designated States except US*): **ONE VOICE TECHNOLOGIES, INC.** [US/US]; Suite 240, 6333 Greenwich Drive, San Diego, CA 92122 (US).
- (72) Inventor; and
- (75) Inventor/Applicant (*for US only*): **WEBER, Dean, C.** [US/US]; 11422 Cypress Terrace Place, San Diego, CA 92131 (US).
- For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*



**WO 01/78065 A1**

(54) Title: NATURAL LANGUAGE AND DIALOGUE GENERATION PROCESSING

(57) Abstract: A system and method for interacting with objects, via a computer using utterances, speech processing and natural language processing. A Dialogue Generation Log provides a conversational memory by relating earlier generated utterances with possible new utterances. The Dialogue Generation Log encompasses a memory structure relating the prior spoken utterances with objects, and conversational contexts. The speech processor searches the Dialogue Generation Log for a matching phrase for the proposed utterance, and determines whether the utterance should be repeated. The determination may be made on a case-by-case basis, depending upon the context of the conversational utterance. Custom user information may be stored globally using a global user registry for remote access. The system reduces the repetitive nature of computer generated conversation by not voicing utterances that have been generated recently, and by learning user preferences from already voiced conversations.

## NATURAL LANGUAGE AND DIALOGUE GENERATION PROCESSING

5

## BACKGROUND OF THE INVENTION

## I. Field of the Invention

The present invention relates to speech recognition for an object-based computer user interface. More specifically, the embodiments relate to a novel method and system for user interaction with a computer using speech recognition and natural language processing.

10

## II. Description of the Related Art

As computers have become more prevalent it has become clear that many people have great difficulty understanding and communicating with computers. A user must often learn archaic commands and non-intuitive procedures in order to operate the computer. For example, most personal computers use windows-based operating systems that are largely menu-driven. This requires that the user learn which menu commands or sequence of commands produce the desired results.

15

20

Furthermore, traditional interaction with a computer is often slowed by manual input devices such as keyboards or mice. Many computer users are not fast typists. As a result, much time is spent communicating commands and words to the computer through these manual input devices. It is becoming clear that an easier, faster and more intuitive method of communicating

with computers and networked objects, such as web-sites, is needed.

One proposed method of computer interaction is speech recognition. Speech recognition involves software and hardware that act together to audibly detect human speech and translate the detected speech into a string of words. As is known in the art, speech recognition works by breaking down sounds the hardware detects into smaller, non-divisible sounds called phonemes. Phonemes are distinct units of sound. For example, the word "those" is made up of three phonemes; the first is the "th" sound, the second is the "o" sound, and the third is the "s" sound. The speech recognition software attempts to match the detected phonemes with known words from a stored dictionary. An example of a speech recognition system is given in U.S. Patent No. 4,783,803, entitled "SPEECH RECOGNITION APPARATUS AND METHOD", issued November 8, 1998, assigned to Dragon Systems, Inc., and incorporated herein by reference. Presently, there are many commercially available speech recognition software packages available from such companies as Dragon Systems, Inc. and International Business Machines, Inc.

One limitation of these speech recognition software packages or systems is that they typically only perform command and control or dictation functions. Thus, the user is still required to learn a vocabulary of commands in order to operate the computer.

A proposed enhancement to these speech recognition systems is to process the detected words using a natural language processing system. Natural language processing generally involves determining a conceptual "meaning" (e.g., what meaning the speaker intended to convey) of the detected words by analyzing their grammatical relationship and relative context. For example, U.S. Patent No. 4,887,212, entitled "PARSER FOR NATURAL LANGUAGE TEXT", issued December 12, 1989, assigned to International Business Machines Corporation

and incorporated by reference herein teaches a method of parsing an input stream of words by using word isolation, morphological analysis, dictionary look-up and grammar analysis.

Natural language processing used in concert with speech recognition provides a powerful tool for operating a computer using spoken words rather than manual input such as a keyboard or mouse. However, one drawback of a conventional natural language processing system is that it may fail to determine the correct "meaning" of the words detected by the speech recognition system. In such a case, the user is typically required to recompose or restate the phrase, with the hope that the natural language processing system will determine the correct "meaning" on subsequent attempts. Clearly, this may lead to substantial delays as the user is required to restate the entire sentence or command. Another drawback of conventional systems is that the processing time required for the speech recognition can be prohibitively long. This is primarily due to the finite speed of the processing resources as compared with the large amount of information to be processed. For example, in many conventional speech recognition programs, the time required to recognize the utterance is long due to the size of the dictionary file being searched.

An additional drawback of conventional speech recognition and natural language processing systems is that they are not interactive, and thus are unable to cope with new situations. When a computer system encounters unknown or new networked objects, new relationships between the computer and the objects are formed. Conventional speech recognition and natural language processing systems are unable to cope with the situations that result from the new relationships posed by previously unknown networked objects. As a result, a conversational-style interaction with the computer is not possible. The user is required to

communicate complete concepts to the computer. The user is not able to speak in sentence fragments because the meaning of these sentence fragments (which is dependent on the meaning of previous utterances) will be lost.

Moreover, conventional speech recognition and natural language processing systems fail to generate intelligent conversations because they lack a conversational memory. In other words, conventional systems do not remember the contents of their conversations with users. As a result, conversations with such system are repetitive, and may result in users repeatedly telling the systems the same facts over-and-over, or worse, have systems fail to learn and remember simple concepts.

Another drawback of conventional speech recognition and natural language processing systems is that once a user successfully "trains" a computer system to recognize the user's speech and voice commands, the user cannot easily move to another computer without having to undergo the process of training the new computer. As a result, changing a user's computer workstations or location results in wasted time by users that need to re-train the new computer to the user's speech habits and voice commands.

What is needed is an interactive user interface for a computer that utilizes speech recognition and natural language processing which avoids the drawbacks mentioned above.

## BRIEF DESCRIPTION OF THE DRAWINGS

The features, objects and advantages of the embodiments discussed herein will become more apparent from the detailed description set forth below when taken in conjunction with the

drawings in which like reference characters identify correspondingly throughout and wherein:

FIG. 1 is a functional block diagram of an exemplary computer system;

FIG. 2 is an expanded functional block diagram of the CPU and storage medium of the computer system of FIG. 1;

5        FIGS. 3A-D are a flowchart of the method of providing interactive speech recognition and natural language processing to a computer;

FIG. 4 is a diagram of selected columns of an exemplary natural language processing (NLP) database;

FIG. 5 is a diagram of an exemplary Database Definition File (DDF);

10        FIG. 6 is a diagram of selected columns of an exemplary object table;

FIGS. 7A-D are a flowchart illustrating the linking of interactive speech recognition and natural language processing to a networked object, such as a web-page;

FIG. 8 is a diagram depicting a computer system connecting to other computers, storage media, and web-sites via the Internet;

15        FIG. 9 is a diagram of an exemplary user global user registry;

FIGS. 10A-B are flowcharts illustrating alternate embodiments of the retrieval and enabling of an individual's user global user registry during login at a computer workstation; and

FIG. 11 is a diagram depicting an exemplary Dialogue Generation Log.

## 20        DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The embodiments discussed herein will now be disclosed with reference to a functional

block diagram of an exemplary computer system 100 of FIG. 1. In FIG. 1, computer system 100 includes a central processing unit (CPU) 102. The CPU 102 may be any general purpose microprocessor or microcontroller as is known in the art, appropriately programmed to perform the method described herein with reference to FIGS. 3A-D. The software for programming the CPU can be found at storage medium 108 or alternatively from another location across a computer network. For example, the CPU 102 may be a conventional microprocessor such as the Pentium III™ processor manufactured by Intel Corporation or the like.

CPU 102 communicates with a plurality of peripheral equipment, including a display 104, manual input 106, storage medium 108, microphone 110, speaker 112, data input port 114 and network interface 116. Display 104 may be a visual display such as a CRT, LCD screen, touch-sensitive screen, or other monitors as are known in the art for visually displaying images and text to a user. Manual input 106 may be a conventional keyboard, keypad, mouse, trackball, or other input device as is known in the art for the manual input of data. Storage medium 108 may be a conventional read/write memory such as a magnetic disk drive, floppy disk drive, CD-ROM drive, silicon memory or other memory device as is known in the art for storing and retrieving data. Significantly, storage medium 108 may be remotely located from CPU 102, and be connected to CPU 102 via a network such as a local area network (LAN), or a wide area network (WAN), or the Internet. Microphone 110 may be any suitable microphone as is known in the art for providing audio signals to CPU 102. Speaker 112 may be any suitable speaker as is known in the art for reproducing audio signals from CPU 102. It is understood that microphone 110 and speaker 112 may include appropriate digital-to-analog and analog-to-digital conversion circuitry as appropriate. Data input port 114 may be any data port as is known in the art for

interfacing with an external accessory using a data protocol such as RS-232, Universal Serial Bus, or the like. Network interface 116 may be any interface as known in the art for communicating or transferring files across a computer network, examples of such networks include TCP/IP, ethernet, or token ring networks. In addition, on some systems, a network  
5 interface 116 may consist of a modem connected to the data input port 114.

Thus, FIG. 1 illustrates the functional elements of a computer system 100. Each of the elements of computer system 100 may be suitable off-the-shelf components as described above. The present embodiment provides a method and system for human interaction with the computer system 100 using speech.

10 As shown in FIG. 8, the computer system 100 may be connected to the Internet 700, a collection of computer networks. To connect to the Internet 700, computer system 100 may use a network interface 116, a modem connected to the data input port 114, or any other method known in the art. Web-sites 710, other computers 720, and storage media 108 may also be connected to the Internet through such methods known in the art.

15 Turning now to FIG. 2, an expanded functional block diagram of CPU 102 and storage medium 108 is illustrated. It is understood that the functional elements of FIG. 2 may be embodied entirely in hardware, firmware, or as software instructions encoded on a computer-readable medium. In the case of a software embodiment, the software may be found at storage medium 108 or at an alternate location across a computer network. CPU 102 includes speech  
20 recognition processor 200, data processor 201, natural language processor 202, and application interface 220. The data processor 201 interfaces with the display 104, storage medium 108, microphone 110, speaker 112, data input port 114, and network interface 116. The data



processor 201 allows the CPU to locate and read data from these sources. Natural language processor 202 further includes variable replacer 204, string formatter 206, word weighter 208, boolean tester 210, pronoun replacer 211, and search engine 213. Storage medium 108 includes a plurality of context-specific grammar files 212, general grammar file 214, dictation grammar 216, context-specific dictation model 217, natural language processor (NLP) database 218, and a dialogue generation log (DGL) file 219. In the preferred embodiment, the grammar files 212, 214, and 216 are Bakus-Naur Form (BNF) files, which describe the structure of the language spoken by the user. BNF files are well known in the art for describing the structure of language, and details of BNF files will therefore not be discussed herein. One advantage of BNF files is that hierarchical tree-like structures may be used to describe phrases or word sequences, without the need to explicitly recite all combinations of these word sequences. Thus, the use of BNF files in the preferred embodiment minimizes the physical sizes of the files 212, 214, and 216 in the storage medium 108, increasing the speed at which these files can be enabled and searched as described below. However, in alternate embodiments, other file structures are used.

The context-specific dictation model 217 is an optional file that contains specific models to improve dictation accuracy. These models enable users to specify word orders and word models. The models accomplish this by describing words and their relationship to other words, thus determining word meaning by contextual interpretation in a specific field or topic. Take for example, the phrase "650 megahertz microprocessor computer." A context-specific dictation model 217 for computers may indicate the likelihood of the word "microprocessor" being associated with "computer," and that a number, such as "650" is likely to be found near the word "megahertz." By interpreting the context of the words, via a context-specific dictation model

217, a speech recognition processor would analyze the phrase, interpret a single object, i.e. the computer, and realize that “650 megahertz microprocessor” are adjectives or traits describing the type of computer.

Topics for context-specific dictation models 217 vary widely, and may include any topic  
5 area of interest to a user—both broad and narrow. Broad topics may include: history, law, medicine, science, technology, or computers. Specialized topics, such as a particular field of literature encountered at a book retailer’s web-site are also possible. Such a context-specific dictation model 217 may contain text for author and title information, for example.

The context-specific dictation model 217 format relies upon the underlying speech  
10 recognition processor 200, and is specific to each type of speech recognition processor 200.

A dialogue generation log 219 is a file, database table, or any other log known in the art that contains a memory of the conversations between a computer or system practicing the present embodiment, and a user of such a system. The purpose behind the dialogue generation log 219 is to create a conversational memory by storing a log of what is spoken. The log is also used to  
15 avoid repetitive speech generation.

An exemplary dialogue generation log 219 is shown in FIG. 11. In this example, the dialogue generation log 219 is comprised of individual dialogue generation log entries 1112A-x, each entry representing a line of dialogue or statement exchanged between the system and a user. Although the format of each entry may vary from embodiment to embodiment, an example  
20 dialogue generation log entry would contain information such as a log entry identifier (ID) 1102, a user identifier 1104, a statement 1106, the context of the statement 1108, and a date/time stamp 1110. The log entry ID 1102 uniquely identifies each entry 1112. A user ID 1104 represents

each user of the system. Statements made by the system itself are also recorded in the log, and a special user ID 1104 is given to the system. In the example illustrated in FIG 11, the system is designated as user "-1," and other users are given positive numbers as their user ID 1104. Other formats to represent users are certainly possible likely. The important part is that the system is uniquely identified with a user ID 1104 different from system users. Next, each entry contains a statement of what was spoken 1106, and preferably the context 1108 in which the statement was spoken. For example, the statement "I would like to see movies," would be linked to the context of "movies" or "film." Lastly, a date-time-stamp 1110 may be recorded with each entry to identify the time frame (i.e. how recent) the statement was last made.

The operation and interaction of these functional elements of FIG. 2 will be described with reference to the flowchart of FIGS. 3A-3D. In FIG. 3A, the flow begins at block 300 with the providing of an utterance to speech processor 200. An utterance is a series of sounds having a beginning and an end, and may include one or more spoken words. A microphone 110 may capture the spoken words in block 300. Alternately, the utterance may be provided to the speech processor 200 over data input port 114, or from storage medium 108. Preferably, the utterance is in a digital format such as the well-known ".wav" audio file format.

The flow proceeds to decision 302 where the speech processor 200 determines whether one of the context-specific grammars 212 has been enabled. If the context-specific grammars 212 are enabled, the context-specific grammars 212 are searched at block 304. In the preferred embodiment, the context-specific grammars 212 are BNF files that contain words and phrases which are related to a parent context. In general, a context is a subject area. For example, in one embodiment applicable to personal computers, examples of contexts may be "news", or

"weather", or "stocks". In such a case, the context-specific grammars 212 would each contain commands, control words, descriptors, qualifiers, or parameters that correspond to a different one of these contexts. The use of contexts provides a hierarchal structure for types of information. Contexts and their use will be described further below with reference to the NLP database 218.

5           If a context-specific grammar 212 has been enabled, the context-specific grammar 212 is searched for a match to the utterance provided at block 300. However, if a context-specific grammar 212 has not been enabled, the flow proceeds to block 308 where the general grammar 214 is enabled.

10           In the preferred embodiment, the general grammar 214 is a BNF file which contains words and phrases which do not, themselves, belong to a parent context, but may have an associated context for which a context-specific grammar file 212 exists. In other words, the words and phrases in the general grammar 214 may be at the root of the hierarchal context structure. For example, in one embodiment applicable to personal computers, the general grammar 214 would contain commands and control phrases.

15           In block 310, the general grammar 214 is searched for a matching word or phrase for the utterance provided at block 300. A decision is made, depending on whether the match is found, at block 312. If a match is not found, then the dictation grammar 216 is enabled at block 314. In the preferred embodiment, the dictation grammar 216 is a BNF file that contains a list of words that do not, themselves, have either a parent context or an associated context. For example, in  
20           one embodiment applicable to a personal computer, the dictation grammar 216 contains a relatively large list of general words similar to a general dictionary.

          In block 316 the dictation grammar is searched for matching words for each word of the

utterance provided at block 300. At decision block 318, if no matching words are found, any relevant context-specific dictation model 217 is enabled at block 317.

At decision block 319, if no matching words are found, a visual error message is optionally displayed at the display 104 or an audible error message is optionally reproduced through speaker 112, at block 320. The process ends until another utterance is provided to the speech processor 200 at block 300.

Thus, as can be seen from the above description, when an utterance is provided to the speech processor 200, the enabled context-specific grammar 212, if any, is first searched. If there are no matches in the enabled context-specific grammar 212, then the general grammar 214 is enabled and searched. If there are no matches in the general grammar 214, then the dictation grammar 316 is enabled and searched. Finally, if there are no matches in the dictation grammar 316, a context-specific dictation model 217 is enabled 317 and used to interpret the utterance.

In the preferred embodiment, when the speech recognition processor 200 is searching either the context-specific grammar 212 or the general grammar 214, it is said to be in the “command and control” mode. In this mode, the speech recognition processor 200 compares the entire utterance as a whole to the entries in the grammar. By contrast, when the speech recognition processor 200 is searching the dictation grammar, it is said to be in the “dictation” mode. In this mode, the speech recognition processor 200 compares the utterance to the entries in the dictation grammar 216 one word at a time. Finally, when the speech recognition processor 200 is matching the utterance with a context-specific dictation model 217, it is said to be in “model matching” mode. It is expected that searching for a match for an entire utterance in the command and control mode will generally be faster than searching for one word at a time in

dictation or model matching modes.

It is further expected that any individual context-specific grammar 212 will be smaller in size (i.e., fewer total words and phrases) than the general grammar 214, which in turn will be smaller in size than the dictation grammar 216. Thus, by searching any enabled context-specific grammar 212 first, it is likely that a match, if any, will be found more quickly, due at least in part to the smaller file size. Likewise, by searching the general grammar 214 before the dictation grammar 216, it is likely that a match, if any, will be found more quickly.

Additionally, as will be explained further below with regard to the ability of the present embodiment to adaptively add to both the context-specific grammar 212 and the general grammar 214, they will contain the most common utterances. As such, it is expected that a match is more likely to be found quickly in the context-specific grammar 212 or the general grammar 214 than in the dictation grammar 216.

Finally, as will be explained further below, the words and phrases in the enabled context-specific grammar 212 are more likely to be uttered by the user because they are words that are highly relevant to the subject matter about which the user was most recently speaking. This also allows the user to speak in a more conversational style, using sentence fragments, with the meaning of his words being interpreted according to the enabled context-specific grammar 212.

By searching in the above-described sequence, the present embodiment may search more efficiently than if the searching were to occur one entry at a time in a single, large list of all expected words and phrases.

Referring back to decision 306, if a match is found in the context-specific grammar 212, then there are three possible next alternatives shown in FIG. 3A. For each matching entry in the

enabled context-specific grammar 212, there may be an associated action to be taken by the speech recognition processor 200. Block 322 shows that one action may be to direct application interface 220 to take some action with respect to a separate software application or entity. For example, application interface 220 may use the Speech Application Programming Interface (SAPI) standard by Microsoft to communicate with an external application. The external application may be directed, for example, to access a particular Internet web site URL or to speak a particular phrase by converting text to speech. Other actions may be taken as will be discussed further below with reference to the NLP database 218 of FIG. 4.

Block 324 shows that another action may be to access a row in the natural language processing (NLP) database 218 directly, thereby bypassing the natural language processing method described further below. Block 326 shows that another action may be to prepend a word or phrase for the enabled context to the matching word or phrase found in the context-specific grammar 306. For example, if the enabled context were "movies" and the matching utterance were "8 o'clock," the word "movies" would be prepended to the phrase "8 o'clock" to form the phrase "movies at 8 o'clock."

Likewise, if a match is found in the general grammar 214, then the flow may proceed to block 322 where the application interface 220 is directed to take an action as described above, or to block 324 where a row in the NLP database is directly accessed. However, if a match is found in the general grammar 214, no prepending of a context occurs because, as stated above, the entries in the general grammar 214 do not, themselves, have a parent context.

Alternatively, manually entered words may be captured, at block 301, and input into the natural language processor.

Finally, with reference to FIG. 3A, words may be entered manually via manual input 106. In this case, no speech recognition is required, and yet natural language processing of the entered words is still desired. Thus, the flow proceeds to FIG. 3B.

In FIG. 3B, at block 328, the natural language processor 202 formats the phrase for  
5 natural language processing analysis. This formatting is accomplished by string formatter 206 and may include such text processing as removing duplicate spaces between words, making all letters lower case (or upper case), expanding contractions (e.g., changing "it's" to "it is"), and the like. The formatting prepares the phrase for parsing.

The flow continues to block 330 where certain "word-variables" are replaced with an  
10 associated wildcard function by variable replacer 204 in preparation for accessing the NLP database 218. As used herein, the term "word-variables" refers to words or phrases that represent amounts, dates, times, currencies, and the like. For example, in one embodiment the phrase "what movies are playing at 8 o'clock" would be transformed at block 330 to "what movies are playing at \$time" where "\$time" is a wildcard function used to represent any time value. As  
15 another example, in one embodiment the phrase "sell IBM stock at 100 dollars" would be transformed at block 330 to "sell IBM stock at \$dollars" where "\$dollars" is a wildcard function used to represent any dollar value. This block may be accomplished by a simple loop that searches the phrase for key tokens such as the words "dollar" or "o'clock" and replaces the word-variables with a specified wildcard function. In order to keep track of the location in the phrase  
20 where the substitution was made, an array may be used. This allows re-substitution of the original word-variable back into the phrase at the some position after the NLP database 218 has been searched.



The purpose of replacing word-variables with an associated wildcard function at block 330 is to reduce the number of entries that must be present in the NLP database 218. For example, the NLP database 218 would only contain the phrase "what movies are playing at \$time" rather than a separate entry for 8 o'clock, 9 o'clock, 10 o'clock, and so on. The NLP database 218 will be described further below.

At block 332, pronouns in the phrase are replaced with proper names by pronoun replacer 211. For example, in one embodiment the pronouns "I," "my," or "mine" would be replaced with the speaker's name. Block 332 allows user-specific facts to be stored and accessed in the NLP database 218. For example, the sentence "who are my children" would be transformed into "who are Dean's children" where "Dean" is the speaker's proper name. Again, this block may be performed in a simple loop that searches the phrase for pronouns, and replaces the pronouns found with an appropriate proper name. In order to keep track of the locations in the phrase where a substitution was made, an array may be used.

In block 334, the individual words in the phrase are weighted according to their relative "importance" or "significance" to the overall meaning of the phrase by word weighter 208. For example, in one embodiment there are three weighting factors assigned. The lowest weighting factor is assigned to words such as "a," "an," "the," and other articles. The highest weighting factor is given to words that are likely to have a significant relation to the meaning of the phrase. For example, these may include all verbs, nouns, adjectives, and proper names in the NLP database 218. A medium weighting factor is given to all other words in the phrase. The purpose of this weighting is to allow for more powerful searching of the NLP database 218.

An example of selected columns of the NLP database 218 of one embodiment is shown in

FIG. 4. The NLP database 218 comprises a plurality of columns 400-410, and a plurality of rows 412A-412N. In column 400, the entries represent phrases that are "known" to the NLP database. In column 402, a number of required words for each entry in column 400 is shown. In column 404, an associated context or subcontext for each entry in column 400 is shown. In columns 408 and 410, one or more associated actions are shown for each entry in column 400. It should be noted that the NLP database 218 shown in FIG. 4 is merely a simplified example for the purpose of teaching the present embodiment. Other embodiments may have more or fewer columns with different entries.

Referring back to FIG. 3B, at block 336, the NLP database 218 is searched for possible matches to the phrase, based on whether the entry in column 400 of the NLP database 218 contains any of the words in the phrase (or their synonyms), and the relative weights of those words. At block 338, a confidence value is generated for each of the possible matching entries based on the number of occurrences of each of the words in the phrase and their relative weights.

Weighted word searching of a database is well known in the art and may be performed by commercially available search engines such as the product "dtsearch" by DT Software, Inc. of Arlington, Virginia. Likewise, searching using synonyms is well known in the art and may be accomplished using such publicly available tools such as "WordNet," developed by the Cognitive Science Laboratory of Princeton University in Princeton, New Jersey. The search engine may be an integral part of the natural language processor 202.

At decision 340, the natural language processor 202 determines whether any of the possible matching entries has a confidence value greater than or equal to some predetermined minimum threshold, T. The threshold T represents the lowest acceptable confidence value for

which a decision can be made as to whether the phrase matched any of the entries in the NLP database 218. If there is no possible matching entry with a confidence value greater than or equal to T, then the flow proceeds to block 342 where an optional error message is either visually displayed to the user over display 104 or audibly reproduced over speaker 112. In one embodiment, the type of error message, if any, displayed to the user may depend on how many “hits” (i.e., how many matching words from the phrase) were found in the highest-confidence NLP database entry. A different type of error message would be generated if there was zero or one hits, than if there were two or more hits.

If, however, there is at least one entry in the NLP database 218 for which a confidence value greater than or equal to T exists, then the flow proceeds to block 344 where the “noise” words are discarded from the phrase. The “noise” words include words that do not contribute significantly to the overall meaning of the phrase relative to the other words in the phrase. These may include articles, pronouns, conjunctions, and words of a similar nature. “Non-noise” words would include words that contribute significantly to the overall meaning of the phrase. “Non-noise” words would include verbs, nouns, adjectives, proper names, and words of a similar nature.

The flow proceeds to FIG. 3C where the non-noise word requirement is retrieved from column 402 of the NLP database 218 for the highest-confidence matching entry at block 346. For example, if the highest-confidence matching phrase was the entry in row 412A, (e.g., “what movies are playing at \$time”), then the number of required non-noise words is 3.

At decision 348, a test is made to determine whether the number of required non-noise words from the phrase is actually present in the highest-confidence entry retrieved from the NLP

database 218. This test is a verification of the accuracy of the relevance-style search performed at block 336, it being understood that an entry may generate a confidence value higher than the minimum threshold, T, without being an acceptable match for the phrase.

5 The nature of the test performed at decision 348 is a boolean "AND" test performed by boolean tester 210. The test determines whether each one of the non-noise words in the phrase (or its synonym) is actually present in the highest-confidence entry. If there are a sufficient number of required words actually present in the highest-confidence entry, then the flow proceeds to block 350, where the natural language processor 202 directs application interface 220 to take an associated action from column 408 or 410. It is understood that additional action  
10 columns may also be present.

For example, if the highest confidence entry was the entry in row 412A, and the boolean test of decision 348 determined that there actually were 3 non-noise words from the phrase in the entry in column 400, then the associated action in column 408 (e.g., access movie web site) would be taken. Other entries in the NLP database have other associated actions. For example,  
15 if the highest-confidence entry is that in row 412E (e.g., "what time is it"), the associated action may be for natural language processor 202 to direct a text-to-speech application (not shown) to speak the present time to the user through the speaker 112. As another example, if the highest-confidence entry is that in row 412F (e.g., "show me the news"), the first associated action may be to access a predetermined news web site on the Internet, and a second associated action may  
20 be to direct an image display application (not shown) to display images associated with the news. Different or additional actions may also be performed.

Also, if the highest-confidence entry contains the required number of non-noise words

from the phrase as determined at decision 348, the natural language processor 202 instructs the speech recognition processor 200 to enable the context-specific grammar 212 for the associated context of column 404. Thus, for row 412A, context-specific grammar 212 for the context "movies" would be enabled. Thus, when the next utterance is provided to the speech recognition processor 200 in block 300 of FIG. 3A, it would search the enabled context-specific grammar 212 for "movies" before searching the general grammar 214. As previously stated, enabling the appropriate context-specific grammar 212 greatly increases the likelihood of fast, successful speech recognition, and enhances the user's ability to communicate with the computer in a conversational style.

If, however, back at decision 348, the required number of non-noise words from the phrase is not actually present in the highest-confidence entry retrieved from the NLP database 218, then the flow proceeds to block 354 where the user is prompted over display 104 or speaker 112 whether the highest-confidence entry was meant, and record the inquiry in the dialogue generation log 219. For example, if the user uttered "How much is IBM stock selling for today," the highest-confidence entry in the NLP database 218 may be the entry in row 412B. In this case, although the relevance factor may be high, the number of required words (or their synonyms) may not be sufficient. Thus, the user would be prompted at block 354 whether he meant "what is the price of IBM stock on August 28, 1998."

The user may respond either affirmatively or negatively. If it is determined at decision 356 that the user has responded affirmatively, then the action associated with the highest-confidence entry is taken at block 350, and the associated context-specific grammar 212 enabled at block 352. The speech made by the user is also recorded as an entry of the dialogue generation

log 219.

If, however, it is determined at decision 356 that the user has responded negatively, then the flow proceeds to FIG. 3D where the associated context from column 404 of NLP database 218 is retrieved for the highest-confidence entry, and the user is prompted for information using a context-based interactive dialog at block 360. For example, if the user uttered “what is the price of XICOR stock today,” and the highest confidence entry from the NLP database 218 was row 412B (e.g., “what is the price of IBM stock on \$date”), then the user would be prompted at block 354 whether that was what he meant.

If the user responds negatively, then the context “stock” is retrieved from column 404 at block 358, and the context-based interactive dialog for the stock context is presented to the user over the display 104 and speaker 112. Such a context-based interactive dialog may entail prompting the user for the name and stock ticker symbol of XICOR stock. The user may respond by speaking the required information. A different context-based interactive dialog may be used for each of the possible contexts. For example, the “weather” context-based interactive dialog may entail prompting the user for the name of the location (e.g., the city) about which weather information is desired. Also, the “news” context-based interactive dialog may entail prompting the user for types of articles, news source, Internet URL for the news site, or other related information.

Upon completion of the context-based interactive dialog, the NLP database 218, general grammar 214, and context-specific grammar 212 are updated to include the new information, at block 362. In this way, the next time the user asks for that information, a proper match will be found, and the appropriate action taken without prompting the user for more information. Thus,

the present embodiment adaptively “learns” to recognize phrases uttered by the user.

In one embodiment, one or more of the NLP database 218, context-specific grammar 212, general grammar 214, and dictation grammar 216 also contain time-stamp values (not shown) associated with each entry. Each time a matching entry is used, the time-stamp value associated with that entry is updated. At periodic intervals, or when initiated by the user, the entries that  
5 have a time-stamp value before a certain date and time are removed from their respective databases/grammars. In this way, the databases/grammars may be kept to an efficient size by “purging” old or out-of-date entries. This also assists in avoiding false matches.

In an alternate embodiment, the updates to the NLP database 218, general grammar 214,  
10 and context-specific grammar 212 are stored in a user global user registry 800, shown in FIG. 9. A global user registry entry 800a would be comprised of any general grammar additions 214a, context-specific grammar additions 212a, and NLP database additions 218a created by the user training. Since each user of the system would have a different global user registry entry 800a, the embodiment would be flexible enough to allow for special customizations and could adapt to the  
15 idiosyncrasies of individual users.

In yet another embodiment, the updates to global user registry entry 800 also contain dialogue generation log additions 219a, user preferences 802, and remembered information 804, as shown in FIG. 9. The dialogue generation log additions 219a would be comprised of the dialogue exchanged between the system and an individual user. The addition of dialogue  
20 generation log additions 219a allow the system greater intelligence in its conversations with an individual user, to be transparent and mobile across different computers. User preferences 802 represents the stored system preferences of an individual user. Examples of such user preferences

802, include, but are not limited to: window color and placement, system sounds, font preferences, and other user-interface preferences. Remembered information 804 includes user schedule information, learned facts derived from the dialogue generation log 219, and any other user (but non-user-interface) stored data not included in user preferences. For example, if the system is told, "I own a dog," the remembered information will include that fact. Remembered information may be stored or represented by any information storage format known in the art, including any database format.

Moreover, in some embodiments, the global user registry 800 would be stored locally and mirrored at known server locations. The mirrored copy, referred to as the "travelling" global user registry entry, enables users to access their phrases "adaptively" learned by the embodiment, even when the user is logged into a different location. FIG. 10A illustrates an embodiment that accesses customized global user registry entries 800a at local and remote (travelling) locations. Initially, a valid system user is verified, by any means known in the art, and then the system searches for a locally stored global user registry entry. For example, the system queries users for their login ID and password as shown in block 900. If the password and login ID match, as determined by decision block 905, the user is deemed to be a valid user. It is well understood that this login ID and password are but one of many methods known in the art to verify valid users, and that all such validation systems could be easily substituted. If no local global user registry entry is found, block 910, the system searches for a travelling global user registry entry, block 920. If either search turns up a global user registry entry, the global user registry entry is loaded, blocks 915 and 925, respectively. Provided that the retrieval of the global user registry entry 800a is successful, blocks 930 and 935, the global user registry entry 800a is enabled at block



940 by extracting the general grammar additions 214a, context-specific grammar additions 212a, NLP database additions 218a, dialogue generation log additions 219a, user preferences 802, and remembered information 804. These “learned” adaptations are then used by the system, as discussed earlier with the method of FIGS 3A-3D. If the retrieval of the travelling global user registry entry 800a is unsuccessful, standard (non-custom user) processing is performed as reflected at block 945.

Alternatively, in some embodiments, the user only logs into the system when accessing a terminal that does not recognize the user as a local user. This embodiment is shown in FIG. 10B. From this remote system, the user is required to specify an ID and a password. Initially, the system determines whether the user has a locally stored voice profile, block 910. If the user has a local voice profile, it is loaded at block 915, and flow continues at decision block 950. If the user does not have a locally stored global user registry entry, the system queries the user for a login identifier and a password at block 900. If the user is valid, the travelling global user registry entry is retrieved at block 925, and flow continues at block 950. Otherwise, the user is treated as a user with standard (non-custom) processing, at block 945. Provided that the retrieval of the global user registry entry 800a is successful, block 950, the global user registry entry 800a is enabled at block 940 by extracting the general grammar additions 214a, context-specific grammar additions 212a, NLP database additions 218a, dialogue generation log additions 219a, user preferences 802, and remembered information 804. These “learned” adaptations are then used by the system, as discussed earlier with the method of FIGS 3A-3D.

In one embodiment, speech recognition and natural language processing may be used to interact with objects, such as help files (“.hlp” files), World-Wide-Web (“WWW” or “web”)

pages, or any other objects that have a context-sensitive voice-based interface.

FIG. 5 illustrates an exemplary Dialog Definition File (DDF) 500 which represents information necessary to associate the speech recognition and natural language processing to an internet object, such as a text or graphics file or, in the preferred embodiment, a web-page or help file. Although in its simplest embodiment the Dialog Definition File 500 consists of an object table 510, the DDF may also contain additional context-specific grammar files 214 and additional entries for the natural language processing (NLP) database 218, as illustrated in FIG.

5 The preferred embodiment of the DDF 500 includes an object table 510, a context-specific grammar file 214, a context-specific dictation model 217, and a file containing entries to the natural language processing database 218. These components may be compressed and combined into the DDF file 500 by any method known in the art, such as through Lempel-Ziv compression.

10 The context-specific specific grammar file 214 and the natural language processing database 218 are as described in earlier sections. The object table 510 is a memory structure, such as a memory tree, chain or table, which associates an address of a resource with various actions, grammars, or entries in the NLP database 218.

15 An exemplary embodiment of the object table 510 is illustrated in FIG. 6. FIG. 6 illustrates a memory table which may contain entry columns for: an object 520, a Text-to-Speech (TTS) flag 522, a text speech 524, a use grammar flag 526, an append grammar flag 528, an "is yes/no?" flag, and "do yes" 532 and "do no" 534 actions. Each row in the table 540A-540n would represent the grammar and speech related to an individual object. The exemplary embodiment would refer to objects 520 through a Universal Resource Locator (URL). A URL is a standard method of specifying the address of any resource on the Internet that is part of the

World-Wide-Web. As this standard is well known in the art for describing the location of Internet resources and objects, the details of URLs will therefore not be discussed herein. One advantage of URLs is that they can specify information in a large variety of object formats, including hypertext, graphical, database and other files, in addition to a number of object devices and communication protocols. However, as shown in FIG. 6, URLs and other method of specifying objects can be used.

When combined with the text speech 524, the Text-to-Speech (TTS) flag 522 indicates whether an initial statement should be voiced over speaker 112 when the corresponding object is transferred. For example, when transferring the web page listed in the object column 520 of row 540A (<http://www.our-pet-store.com>), the TTS flag 522 is marked, indicating the text speech 524, "Hello, welcome to...", is to be voiced over speaker 112.

In some embodiments, the utterance of the initial statement may be dependent on the dialogue generation log 219. In these embodiments, the system first checks to see if the statement was made recently to a user, before re-voicing the statement. There are several advantages in doing this. By consulting the dialogue generation log 219, a conversation can be made more intelligent by using the dialogue generation log 219 as a "conversational memory." Moreover, a conversational memory formed by the dialogue generation log 219 also acts as a "sanity check" to make sure conversations do not become repetitive.

The use of the dialogue generation log 219 is best illustrated by example. Suppose a user was examining a speech-enhanced pet-store web-site using the present embodiment. Initially, an object table 510 is transferred from the web-site to the user's computer system 100. When transferring the web page listed in the object column 520, the system notes that the TTS flag 522

is marked, indicating the text speech 524, "Hello, welcome to our pet store, would you like to see our dog specials?" Since this is the user's first visit to the web page, no context or statement-related entries are found in the dialogue generation log 219, so the statement is to be voiced over speaker 112. The user responds, "no," and decides to surf a different web page instead.

5 Transparent to the user, the system records both statements in the dialogue generation log 219. Moments later, the user presses the web-browser "back" key, returning to the original pet-store web page. At this time, the user returns to the web page, and again, the system notes that the TTS flag 522 is marked, indicating the text speech 524, "Hello, welcome to our pet store, would you like to see our dog specials?" Since this is the user's subsequent visit to the web page, a  
10 statement-related entry is found in the dialogue generation log 219. Since statement was recently voiced over the speaker, it is not voiced on again on a subsequent visit.

The next three object table 510 flags relate to the use of grammars associated with this object. The affirmative marking of the "use grammar" 526 or "append grammar" 528 flags indicate the presence of a content-specific grammar file 214 related to the indicated object. The  
15 marking of the "use grammar" flag 526 indicates that the new content-specific grammar file 214 replaces the existing content-specific grammar file, and the existing file is disabled. The "append grammar" flag 528 indicates that the new content-specific grammar file should be enabled concurrently with the existing content-specific grammar file.

Lastly, the remaining column entries relate to a "yes/no" grammar structure. If the "Is  
20 yes/no?" flag 530 is marked, then a standard "yes/no" grammar is enabled. When a standard "yes/no" grammar is enabled, affirmative commands spoken to the computer result in the computer executing the command indicated in the "Do Yes" entry 532. Similarly, a negative

command spoken to the computer results in the computer executing the command indicated in the "Do No" entry 534. The entries in the "Do Yes" 532 and "Do No" 534 columns may either be commands or pointers to commands imbedded in the NLP Database 218. For example, as shown in row 540B, the "Is Yes/No?" flag is marked. An affirmative answer, such as "yes,"  
5 given to the computer, would result in executing the corresponding command in the "Do Yes" entry 532; in this specific case, the entry is the number "210," a reference to the 210<sup>th</sup> command in the NLP database. An answer of "no" would result in the computer executing the 211<sup>th</sup> command in the NLP database.

Turning now to FIG. 7A, a method and system of providing speech and voice commands  
10 to objects, such as a computer reading a help file or browsing the World-Wide-Web, is illustrated. The method of FIGS. 7A-7D may be used in conjunction with the method of FIGS 3A-3D and FIG. 10. In block 602, an object location is provided to a help file reader or World-Wide-Web browser. A help file reader/browser is a program used to examine hypertext documents that are written to help users accomplish tasks or solve problems, and is well known  
15 in the art. The web browser is a program used to navigate through the Internet, and is well known in the art. Block 602, which providing an object location to the browser, can be as simple as a user clicking on a program "help" menu item, manually typing in a URL, or having a user select a "link" at a chosen web-site. It also may be the result of a voiced command as described earlier with reference to the action associated with each entry in the NLP database 218. Given  
20 the object location, the computer must decide on whether it can resolve object location specified, at block 604. This resolution process is a process well known in the art. If the computer is unable to resolve the object location or internet address, an error message is displayed in the

browser window, at block 605, and the system is returned to its initial starting state 600. If the object location or Internet address is resolved, the computer retrieves the object at block 606. For a networked object, for example, a web browser sends the web-site a request to for the web page, at block 606. For a help file application, the help reader reads the help file off of storage media 108, at block 606.

A decision is made, depending upon whether the object is retrieved, at block 608. If the object cannot be retrieved, an error message is displayed in the browser window, at block 605, and the system is returned to its initial starting state 600. If the object is retrieved, it is displayed in the help-reader or web-site browser, as appropriate, at block 610.

In decision block 612, the computer 100 determines whether the DDF file 500 corresponding to the object is already present on the computer 100. If the DDF file is present, the flow proceeds to FIG. 7C, if not the flow proceeds to FIG. 7B.

Moving to FIG. 7B, if the DDF file 500 is not present, the computer examines whether the DDF file 500 location is encoded within the object. For example, the DDF file location could be encoded within web page HyperText Markup Language (HTML) as a URL. (Note that HTML is well known in the art, and the details of the language will therefore not be discussed herein.) Encoding DDF file location within HTML code may be done either through listing the DDF file location in an initial HTML meta-tag such as:

```
<meta DDF= "http://www.our-pet-store.com/ConverseIt.ddf">
```

or directly through a scripting tag written into the variation of HTML supported by the browser,

```
<!--
```

```
<DDF= "http://www.our-pet-store.com/ConverseIt.ddf">
```

```
-->
```

If the DDF file location information is encoded within the web page, the location's Internet address is resolved, at block 616, and the computer requests transfer of the DDF file 500, at block 626. An equivalent encoding scheme could be used within help file hypertext.

Alternatively, if the DDF file 500 location is not encoded within the object, there are  
5 several alternate places that it may be stored. It may be stored in a pre-defined location at a web-site, such as a certain file location in the root directory, or at a different centralized location, such as another Internet server or the storage medium 108 of FIG. 1. Blocks 618 and 620 test for these possibilities. Block 618 determines whether the DDF file is located at the web-site. At block 618, the computer sends query to the web-site inquiring about the presence of the DDF file  
10 500. If the DDF file 500 is present at the web-site, the computer requests transfer of the DDF file 500, at block 626. If the DDF file 500 is not located at the web-site, the computer queries the centralized location about the presence of a DDF file for the web-site, at block 620. If the DDF file is present at the web-site, the computer requests transfer of the DDF file, at block 626. If the DDF file 500 cannot be found, the existing components of any present DDF file, such as the  
15 object table 510, context-specific dictation model 217, NLP database 218 associated with the object, and context-specific grammar 214 for any previously-viewed object, are deactivated in block 622. Furthermore, the object is treated as a non-voice-activated object, and only standard grammar files are used, at block 624. Standard grammar files are the grammar files existing on the system excluding any grammars associated with the content-specific grammar file associated  
20 with the object.

If the DDF file 500 is requested at block 626, and its transfer is unsuccessful, any existing components of any present DDF file 500 are deactivated, at block 622, and the web-site is treated

as a non-voice-activated object, and only standard grammar files are used, at block 624.

If the DDF file 500 is requested at block 626 and its transfer is successful at block 628, it replaces any prior DDF file, at block 630. Any components of the DDF file 500, such as the object table 510, context-specific-grammar files 214, context-specific-dictation models 217, and  
5 NLP database 218 are extracted at block 632. A similar technique may be used for obtaining the software necessary to implement the method illustrated in FIGS. 3A-3D, comprising the functional elements of FIG. 2.

The flow moves to FIG. 7C. The object table 510 is read into memory by the computer in block 634. If the object is present in the site object table 510, as determined by block 636, it  
10 will be represented by a row 540A-540n of the table, as shown in FIG. 6. Each row of the object table represents the speech-interactions available to a user for that particular object. If no row corresponding to the object exists, then no-speech interaction exists for the web page, and processing ends.

If the object location is present in the site object table 510, as determined by block 636,  
15 the computer checks if the TTS flag 522 is marked, to determine whether a text speech 524 is associated with the web-page, at block 638. If there is a text speech 524, a decision must be made on whether it is appropriate to play the spoken statement 639. For example, if a user was queried, "Would you like to hear about the history of Mars?" in the past five minutes, it would be inappropriate, and probably annoying, to ask the user the same question again. In making such a  
20 determination, the system checks whether the statement has been made to the user recently by examining the dialogue generation log 219.

An individual user may be matched with a statement or statement context within a recent



interval. Suppose the system asked, "Do you own a cat?" and the user replied, "No, I hate cats." The conversation would be recorded in the dialogue generation log 219, noting that the context of the conversation is about cats. On a subsequent visit to a pet-store web-site, the system notes that the TTS flag 522 is marked, indicating the text speech 524, "Hello, welcome to our pet store, would you like to see our cat specials?" Since the text-to-speech statement is an inquiry about cats, the system performs a check in the dialogue generation log 219 to see if there are past conversations with a context 1108 of "cat." The context of "cat" is located, and the system notes that the user responded "no," therefore the user does not own a cat. Consequently, the system realizes that a negative match has been made, and it would be inappropriate to voice the question to the user.

Thus, since dialogue generation log entries 1112 are time dependent, they must be kept up-to-date by the system, and old entries should be culled and removed on a periodic basis. After all, in the case of the above example, it is possible that the user eventually acquires a cat. Old entries may be removed after a certain time-period expires, or after the dialogue generation log 219 reaches a specific size. Dialogue generation log entries 1112 for each user can be removed on a first-in-first-out (FIFO) basis, so that older entries are removed before more recent entries.

If it is appropriate, the statement is voiced at block 640, and flow continues on to FIG 7D. If there is a context-specific grammar file associated with the object, as determined by decision block 642, it is enabled at block 644, and then the NLP database 218 is enabled at block 646. If no context-specific grammar file is associated with the object, only the NLP database 218 is enabled at block 646. Once the NLP database is enabled 646, the system behaves as shown in FIG. 3A-3C, as described above.

In summary, the present embodiment provides a method and system for an object interactive user-interface for a computer. By the use of context-specific grammars that are tied to internet-objects through a Dialog Definition File, the present embodiment decreases speech recognition time and increases the user's ability to communicate with local and networked objects, such as help files or web-pages, in a conversational style. Adaptive updating of the various grammars and the NLP database, the present embodiment further increases interactive efficiency. The adaptive updates can be incorporated into global user registry entries that can be stored locally and remotely, to allow users access to the global user registry entries at various locations.

The previous description of the embodiments is provided to enable any person skilled in the art to make or use the present invention. The various modifications to these embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without the use of inventive faculty. Thus, the present invention is not intended to be limited to the embodiments shown herein, but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

I CLAIM:

## CLAIMS

- 1 1. A method for interacting with an object via a computer using utterances, the method  
2 comprising:  
3 retrieving an object table associated with the object;  
4 searching the object table for a matching entry for the object;  
5 locating an action associated with the matching entry, if the matching entry is found in the  
6 object table;  
7 determining whether the action associated with the matching entry is appropriate;  
8 performing the action associated with the matching entry, if the action is deemed  
9 appropriate.
- 1 2. The method of claim 1 wherein the action associated with the matching entry is voicing a  
2 text statement.
- 1 3. The method of claim 2 wherein the process of determining whether the action associated  
2 with the matching entry is appropriate further comprises:  
3 identifying a context of the voiced text statement associated with the matching entry;  
4 searching a log containing log entries with context and timestamp information to locate  
5 matching log entries with the same context as the voiced text statement;  
6 determining whether the voiced text statement is appropriate based on the timestamp of the  
7 matching log entries.
- 1 4. The method of claim 2 wherein the process of determining whether the action associated  
2 with the matching entry is appropriate further comprises:

3           searching a log containing log entries with statement and timestamp information to locate  
4 matching log entries with the same voiced text statement;

5           determining whether the voiced text statement is appropriate based on the timestamp of the  
6 matching log entries.

1   5.     The method of claim 3 or 4 wherein the object table is read from a web-site.

1   6.     The method of claim 3 or 4 wherein the object table is read from a location that stores  
2 object tables for multiple objects.

1   7.     A method for interacting with an object via a computer using utterances, the method  
2 comprising:

3           searching a context-specific grammar file for a matching phrase for the utterance;

4           searching a general grammar file for the matching phrase if the matching phrase is not  
5 found in the context-specific grammar file;

6           searching a dictation grammar for the matching phrase if the matching phrase is not found  
7 in the general grammar file;

8           searching a context-specific dictation model for the matching phrase if the matching phrase  
9 is not found in the dictation grammar file;

10          searching a database for a matching entry for the matching phrase; and

11          performing an action associated with the matching entry and saving the utterances to a log  
12 file if the matching entry is found in the database.

1   8.     The method of claim 7 the log is a dialogue generation log.

1   9.     The method of claim 8 further comprising:

2           replacing at least one word in the matching phrase prior to the searching the database.

- 1 10. The method of claim 9 wherein the replacing the at least one word comprises substituting a  
2 wildcard for the at least one word.
- 1 11. The method of claim 10 wherein the replacing at least one word comprises substituting a  
2 proper name for the at least one word.
- 1 12. The method of claim 11 further comprising:  
2 text formatting the matching phrase prior to the searching of the database.
- 1 13. The method of claim 12 further comprising:  
2 weighting individual words in the matching phrase according to a relative significance of  
3 the individual words prior to the searching of the database.
- 1 14. The method of claim 13 further comprising:  
2 updating a global user registry entry with the dialogue generation log with utterances  
3 spoken by a user.
- 1 15. The method of claim 13 further comprising:  
2 updating the global user registry entry with at least one of the database, the context-specific  
3 grammar file and the general grammar file with the matching phrase if the matching entry is not  
4 found in the database.
- 1 16. The method of claim 15 further comprising:  
2 generating a confidence values for the matching entry;  
3 comparing the confidence value with a threshold value;  
4 determining whether a required number of words from the matching phrase are present in  
5 the matching entry;  
6 determining the utterance has been spoken before by examining the dialogue generation

7 log; and

8 prompting a user whether the matching entry is a correct interpretation of the utterance if  
9 the required number of words from the matching phrase are not present in the matching entry, nor  
10 present in the dialogue generation log.

1 17. The method of claim 16 further comprising:

2 prompting a user for additional information if the matching entry is not a correct  
3 interpretation of the utterance.

1 18. The method of claim 17 further comprising:

2 updating at least one of the database, the context-specific grammar file, the general  
3 grammar file, and the dialogue generation log with the additional information.

1 19. A system for interacting with a computer using utterances, the system comprising:

2 a speech processor for searching a context-specific grammar file for a matching phrase for  
3 the utterance, for searching a general grammar file for the matching phrase if the matching phrase is  
4 not found in the context-specific grammar file, for searching a dictation grammar for the matching  
5 phrase if the matching phrase is not found in the general grammar file, and for searching a context-  
6 specific dictation model if the matching phrase is not found in the dictation grammar;

7 a natural language processor for searching a database for a matching entry for the matching  
8 phrase and for searching a dialogue generation log for a matching log entry for the utterance; and

9 an application interface for performing an action associated with the matching entry if the  
10 matching entry is found in the database and the action is appropriate if the matching log entry is  
11 found in the dialogue generation log.

1 20. A computer readable medium encoded with executable program code, the code comprising:

2           retrieving an object table associated with the object;  
3           searching the object table for a matching entry for the object;  
4           locating an action associated with the matching entry, if the matching entry is found in the  
5 object table;  
6           determining whether the action associated with the matching entry is appropriate;  
7           performing the action associated with the matching entry, if the action is deemed  
8 appropriate.

1   21.    The computer readable medium of claim 20 wherein the action associated with the  
2 matching entry is voicing a text statement.

1   22.    The computer readable medium of claim 21 wherein the process of determining whether the  
2 action associated with the matching entry is appropriate further comprises:

3           identifying a context of the voiced text statement associated with the matching entry;  
4           searching a log containing log entries with context and timestamp information to locate  
5 matching log entries with the same context as the voiced text statement;  
6           determining whether the voiced text statement is appropriate based on the timestamp of the  
7 matching log entries.

1   23.    The computer readable medium of claim 21 wherein the process of determining whether the  
2 action associated with the matching entry is appropriate further comprises:

3           searching a log containing log entries with statement and timestamp information to locate  
4 matching log entries with the same voiced text statement;  
5           determining whether the voiced text statement is appropriate based on the timestamp of the  
6 matching log entries.

1 24. The computer readable medium of claim 22 or 23 wherein the object table is read from a  
2 web-site.

1 25. The computer readable medium of claim 22 or 23 wherein the object table is read from a  
2 location that stores object tables for multiple objects.

1 26. A computer readable medium encoded with executable program code, the code  
2 comprising:

3 searching a context-specific grammar file for a matching phrase for the utterance;

4 searching a general grammar file for the matching phrase if the matching phrase is not  
5 found in the context-specific grammar file;

6 searching a dictation grammar for the matching phrase if the matching phrase is not found  
7 in the general grammar file;

8 searching a context-specific dictation model for the matching phrase if the matching phrase  
9 is not found in the dictation grammar file;

10 searching a database for a matching entry for the matching phrase; and

11 performing an action associated with the matching entry and saving the utterances to a log  
12 file if the matching entry is found in the database.

1 27. The computer readable medium of claim 26 wherein the log is a dialogue generation log.

1 28. The computer readable medium of claim 27 further comprising:

2 replacing at least one word in the matching phrase prior to the searching the database.

1 29. The computer readable medium of claim 28 wherein the replacing the at least one word  
2 comprises substituting a wildcard for the at least one word.

1 30. The computer readable medium of claim 29 wherein the replacing at least one word



2 comprises substituting a proper name for the at least one word.

1 31. The computer readable medium of claim 30 further comprising:

2 text formatting the matching phrase prior to the searching of the database.

1 32. The computer readable medium of claim 31 further comprising:

2 weighting individual words in the matching phrase according to a relative significance of  
3 the individual words prior to the searching of the database.

1 33. The computer readable medium of claim 32 further comprising:

2 updating a global user registry entry with the dialogue generation log with utterances  
3 spoken by a user.

1 34. The computer readable medium of claim 32 further comprising:

2 updating the global user registry entry with at least one of the database, the context-specific  
3 grammar file and the general grammar file with the matching phrase if the matching entry is not  
4 found in the database.

1 35. The computer readable medium of claim 34 further comprising:

2 generating a confidence values for the matching entry;

3 comparing the confidence value with a threshold value;

4 determining whether a required number of words from the matching phrase are present in  
5 the matching entry;

6 determining the utterance has been spoken before by examining the dialogue generation  
7 log; and

8 prompting a user whether the matching entry is a correct interpretation of the utterance if  
9 the required number of words from the matching phrase are not present in the matching entry, nor

10 present in the dialogue generation log.

1 36. The computer readable medium of claim 35 further comprising:

2 prompting a user for additional information if the matching entry is not a correct  
3 interpretation of the utterance.

1 37. The computer readable medium of claim 36 further comprising:

2 updating at least one of the database, the context-specific grammar file, the general  
3 grammar file, and the dialogue generation log with the additional information.

100


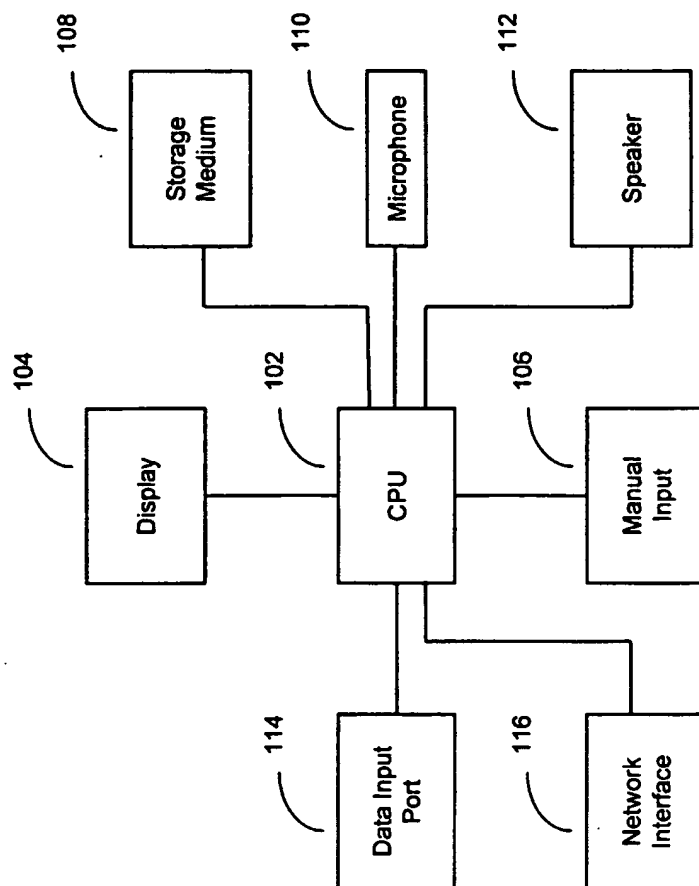



FIG. 1

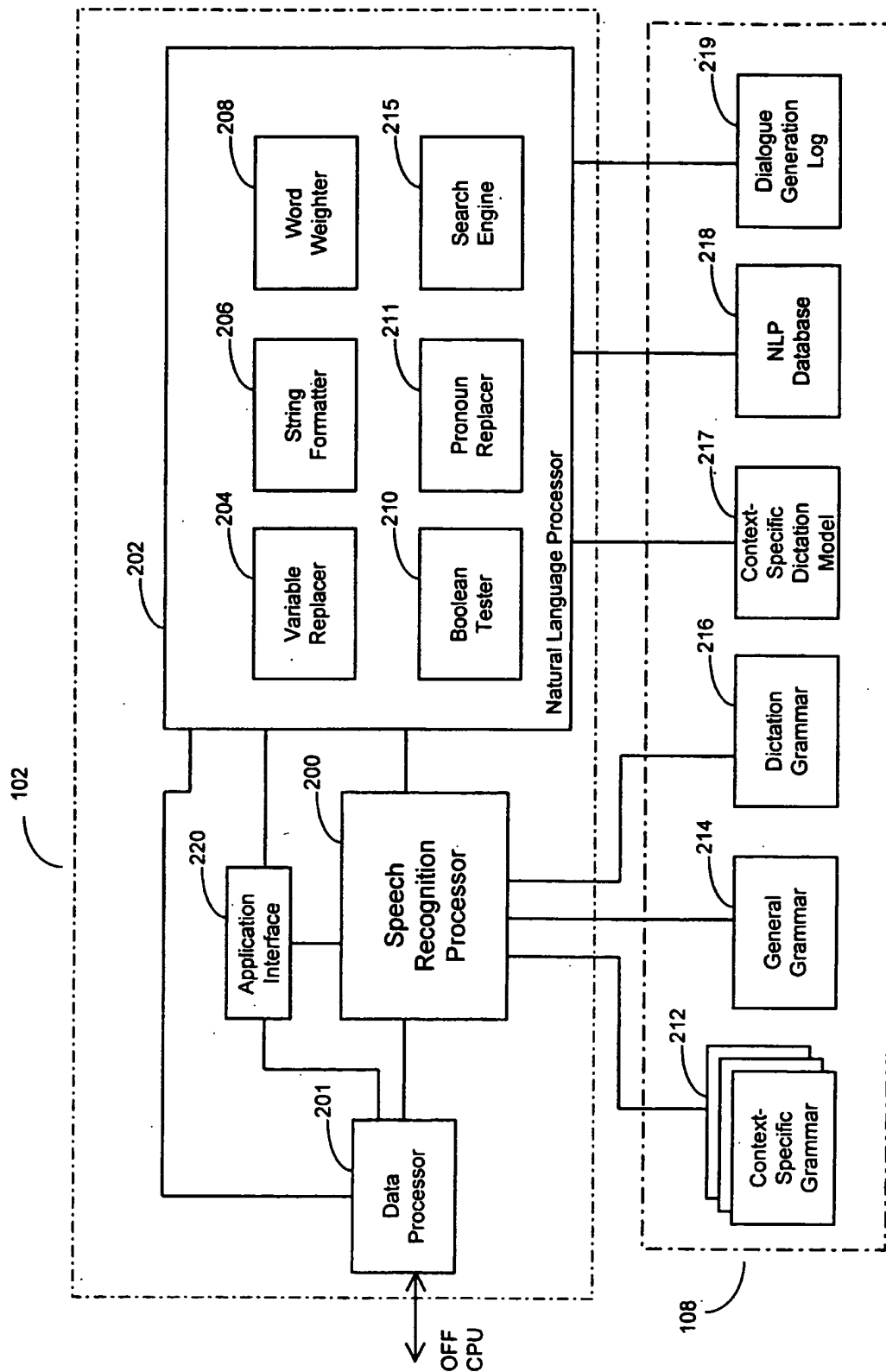


FIG. 2

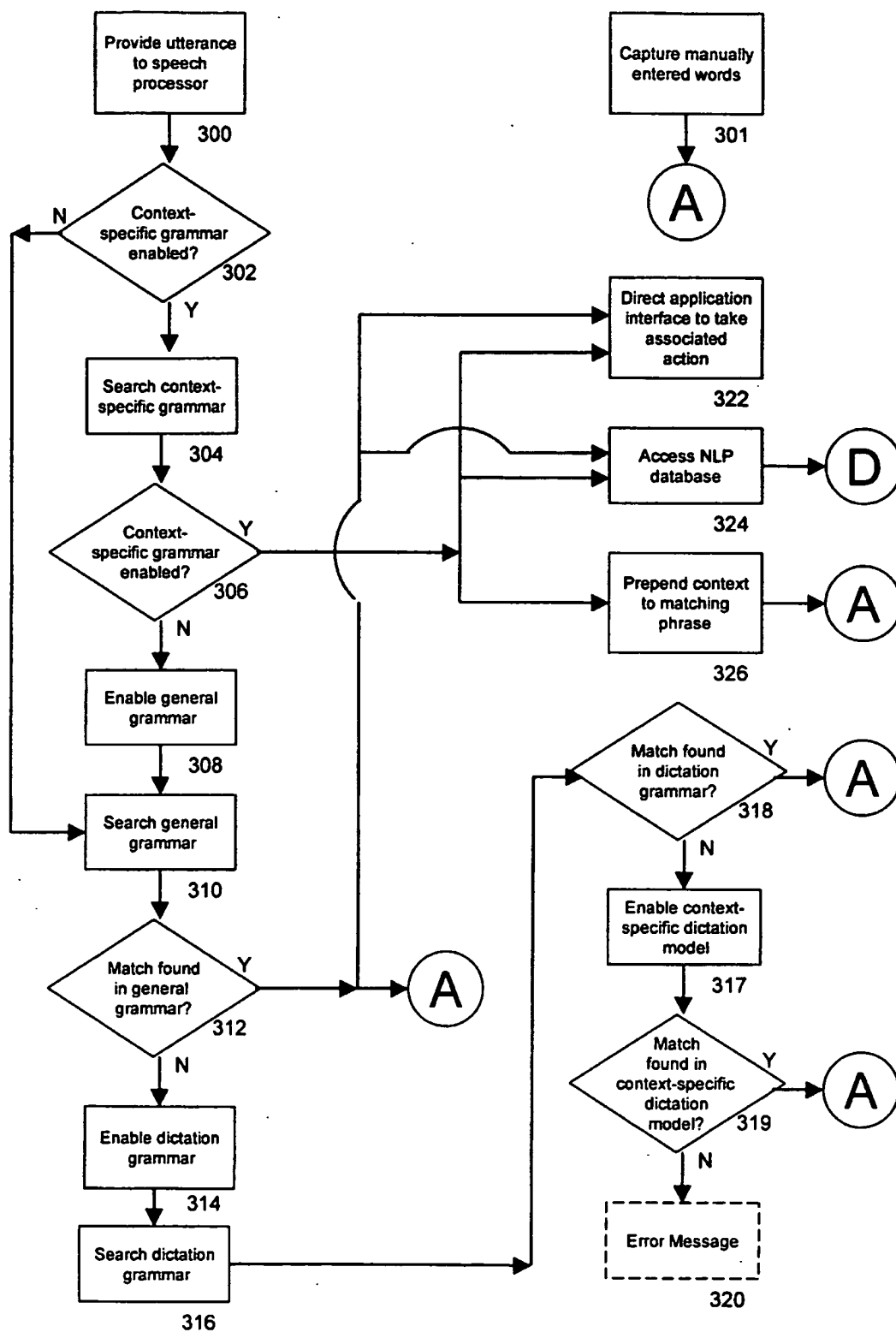


FIG. 3A

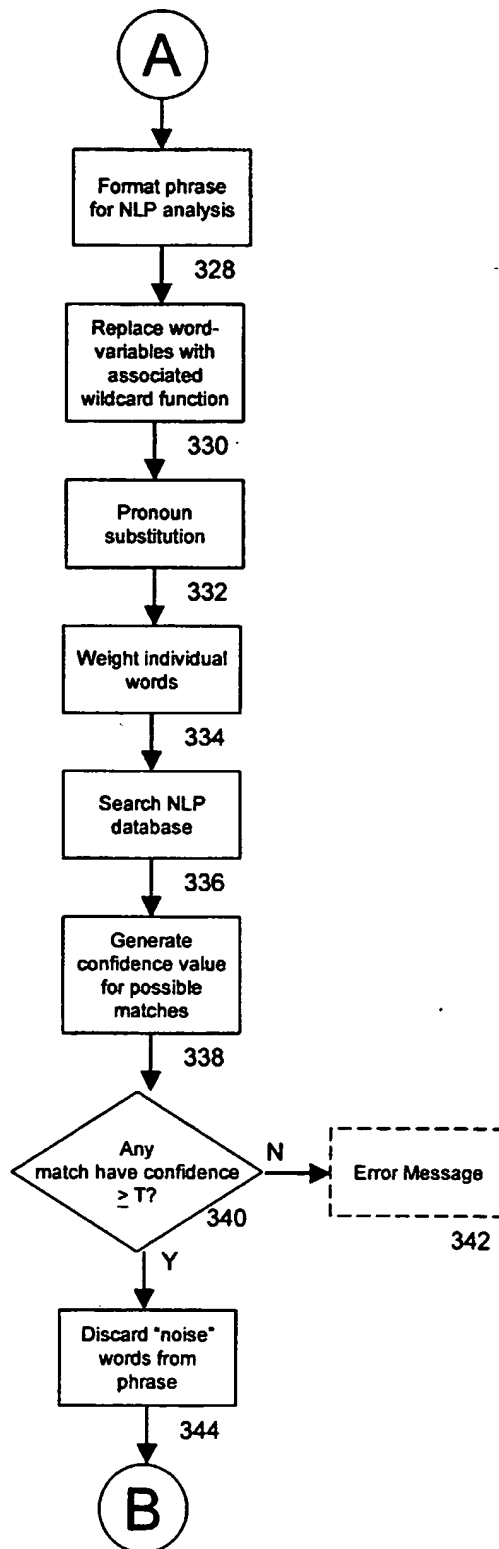


FIG. 3B

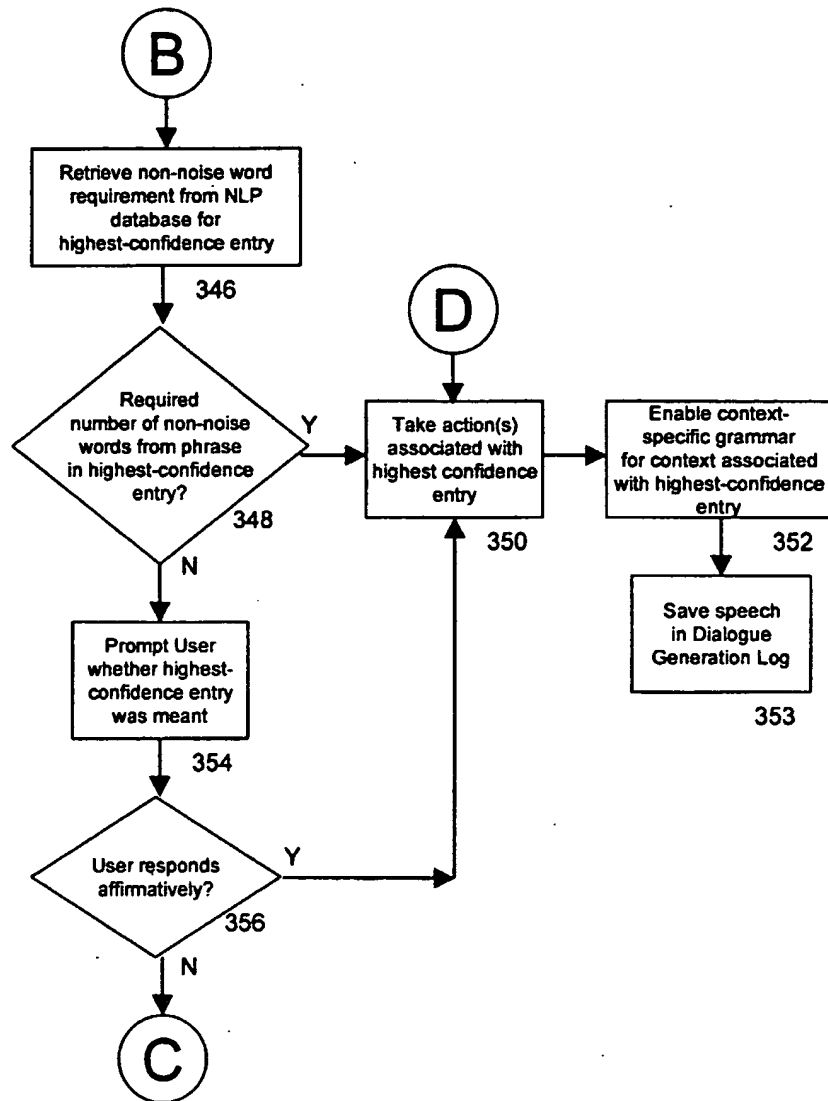


FIG. 3C

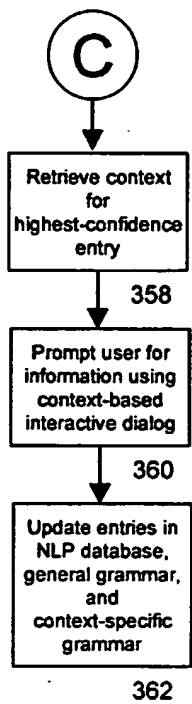



FIG. 3D



218



Phrase	Required words	Context/Subcontext	Action 1	Action 2
What movies are playing at \$time	3	movies	Access movie web site	N/A
What is the price of IBM stock on \$date	4	stocks	Access stock price web site	N/A
Sell IBM stock at \$dollars	4	stocks	Access stock price web site	N/A
What is the weather at \$location	3	weather	Access weather web site	N/A
What time is it	2	time	N/A	Text-to-Speech of Time
Show me the news	2	news	Access news web site	Display Images
How do I format this paragraph	2	Word Processor	Locate Word Processor Help	Format Paragraph Help
		...		
How do I insert a table	2	Spreadsheet	Locate Spreadsheet Help	Insert Table Help

FIG. 4

500

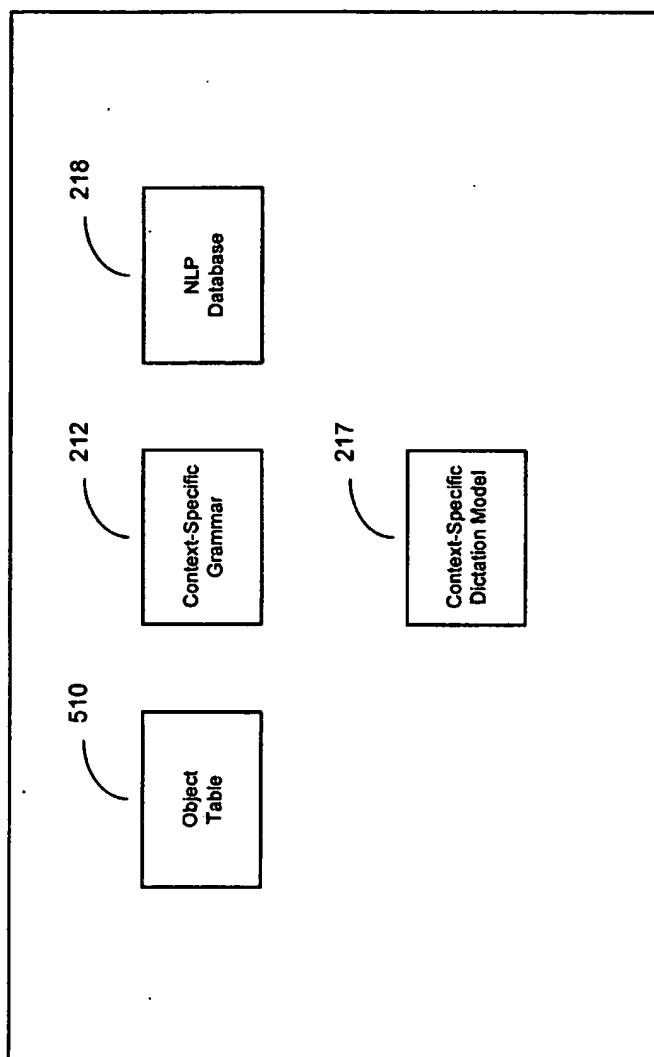



FIG. 5



object	TTS Flag	Text Speech	Use Grammar	Append Grammar	Is Yes/No?	Do Yes	Do No
http://www.our-pet-store.com	x	"Hello, welcome to ...		x	x	200	205
http://www.our-pet-store.com/inquiry	x	"Would you like to learn...			x	210	211
http://www.our-pet-store.com/products	x	"Our current sale is ...	x				
http://www.our-pet-store.com/dogs	x	"Good news dog lovers!...					
http://www.our-pet-store.com/cats	x	"Cat products...			x	254	213
WPHelp:Format:Paragraph	x	"To format paragraphs...					
⋮							
SpreadsheetHelp:Insert:Table		x	"Inserting Tables is easy..				

540A-n

FIG. 6

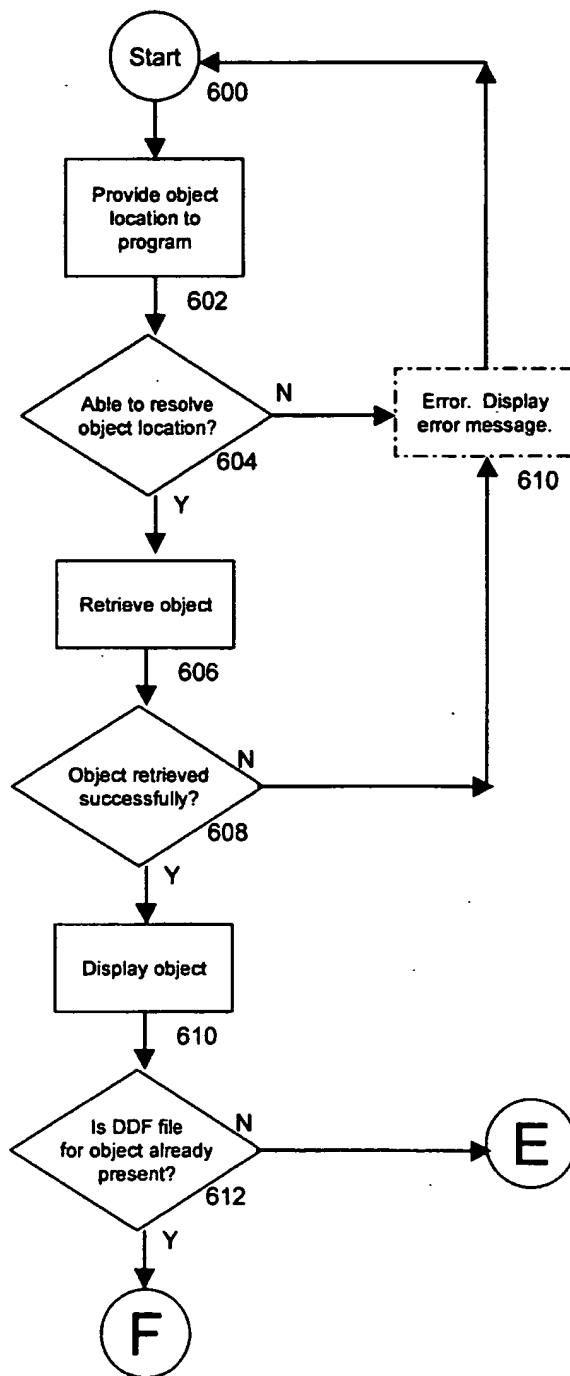


FIG. 7A

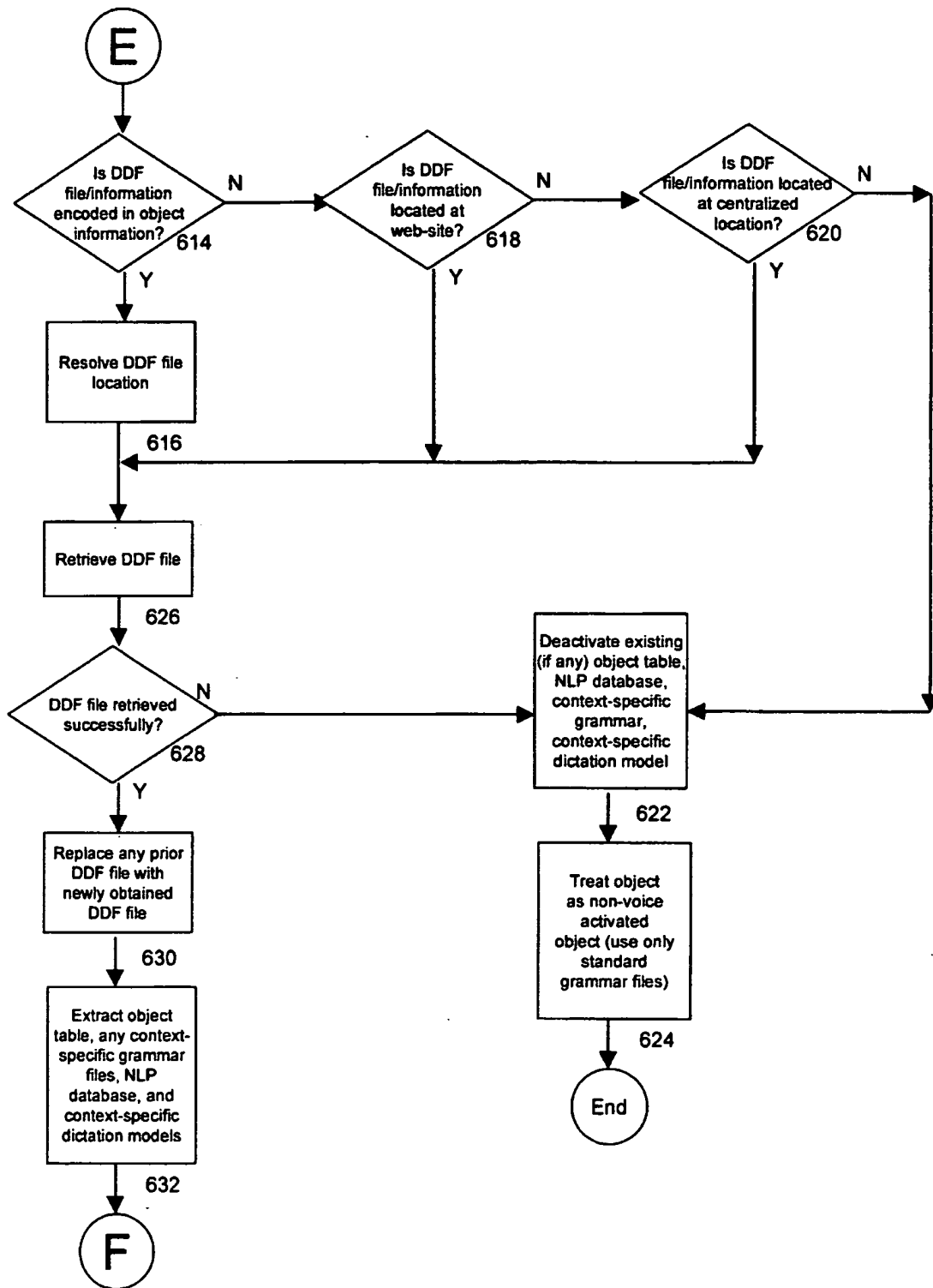


FIG. 7B

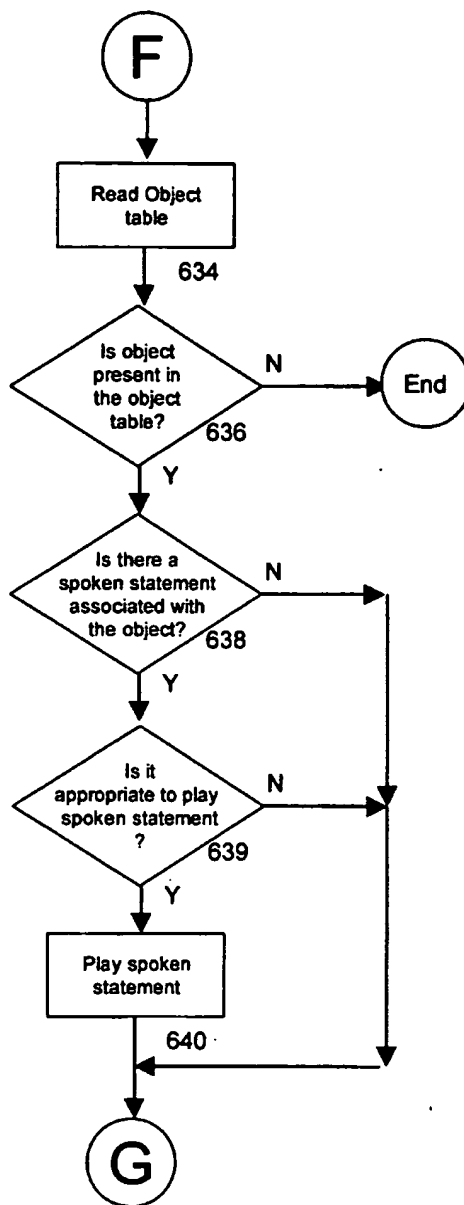


FIG. 7C

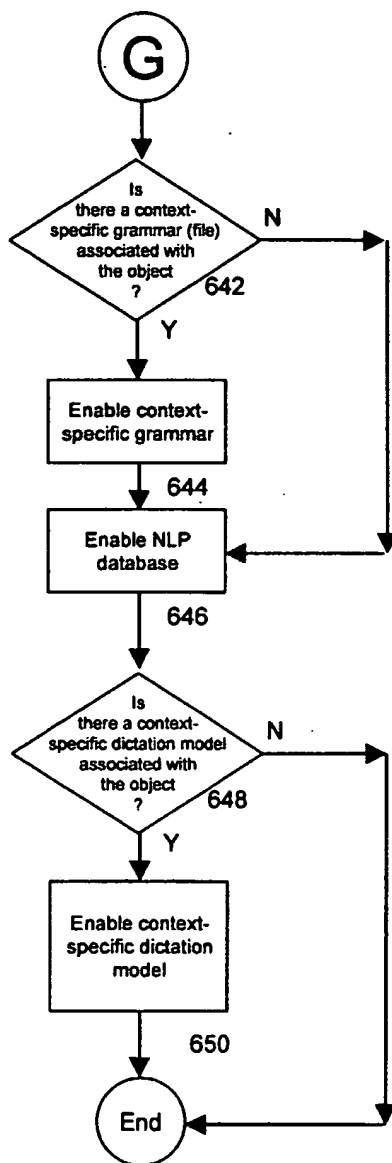


FIG. 7D

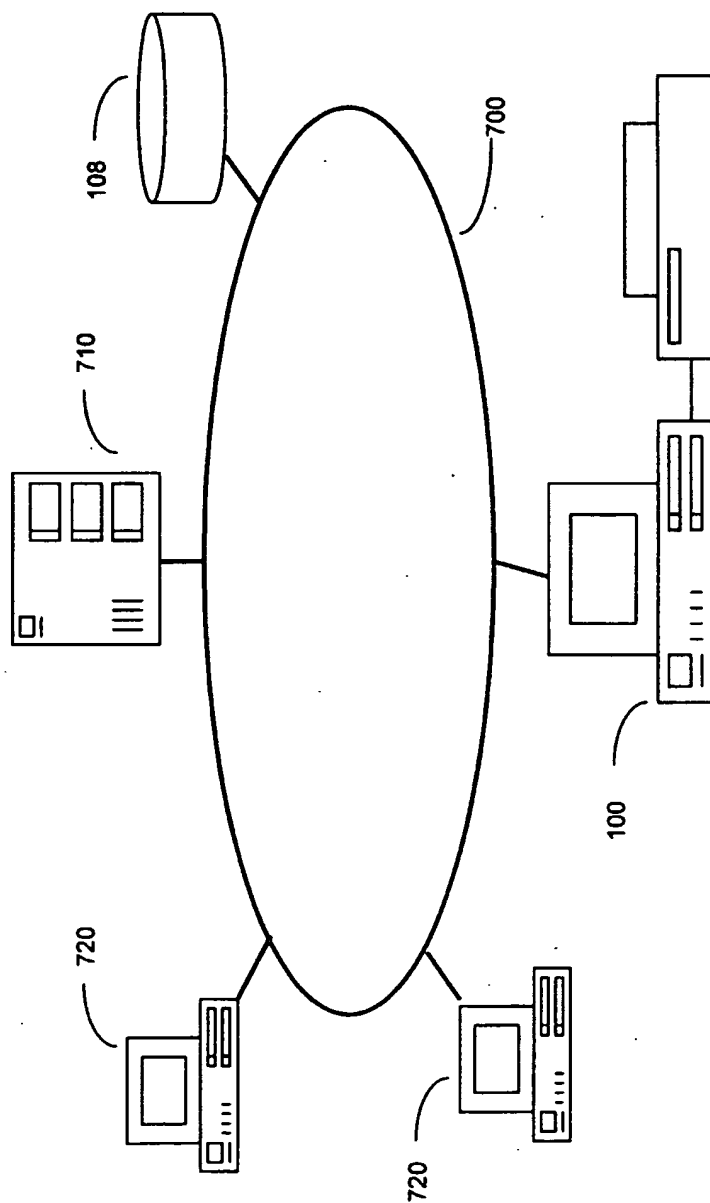


FIG. 8



800

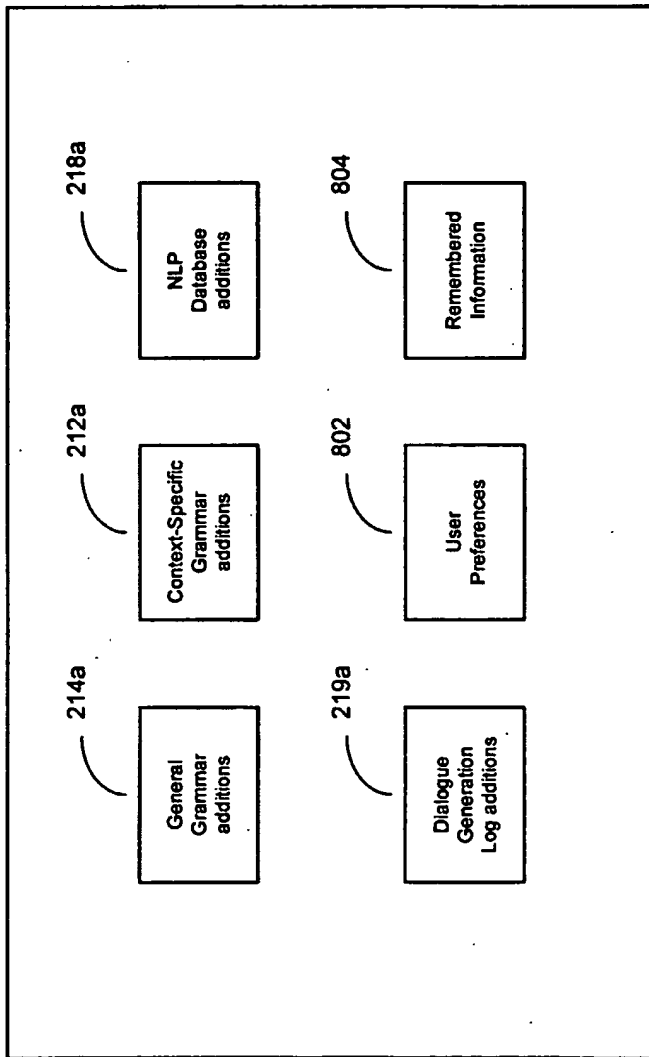
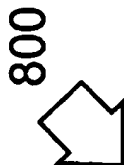


FIG. 9

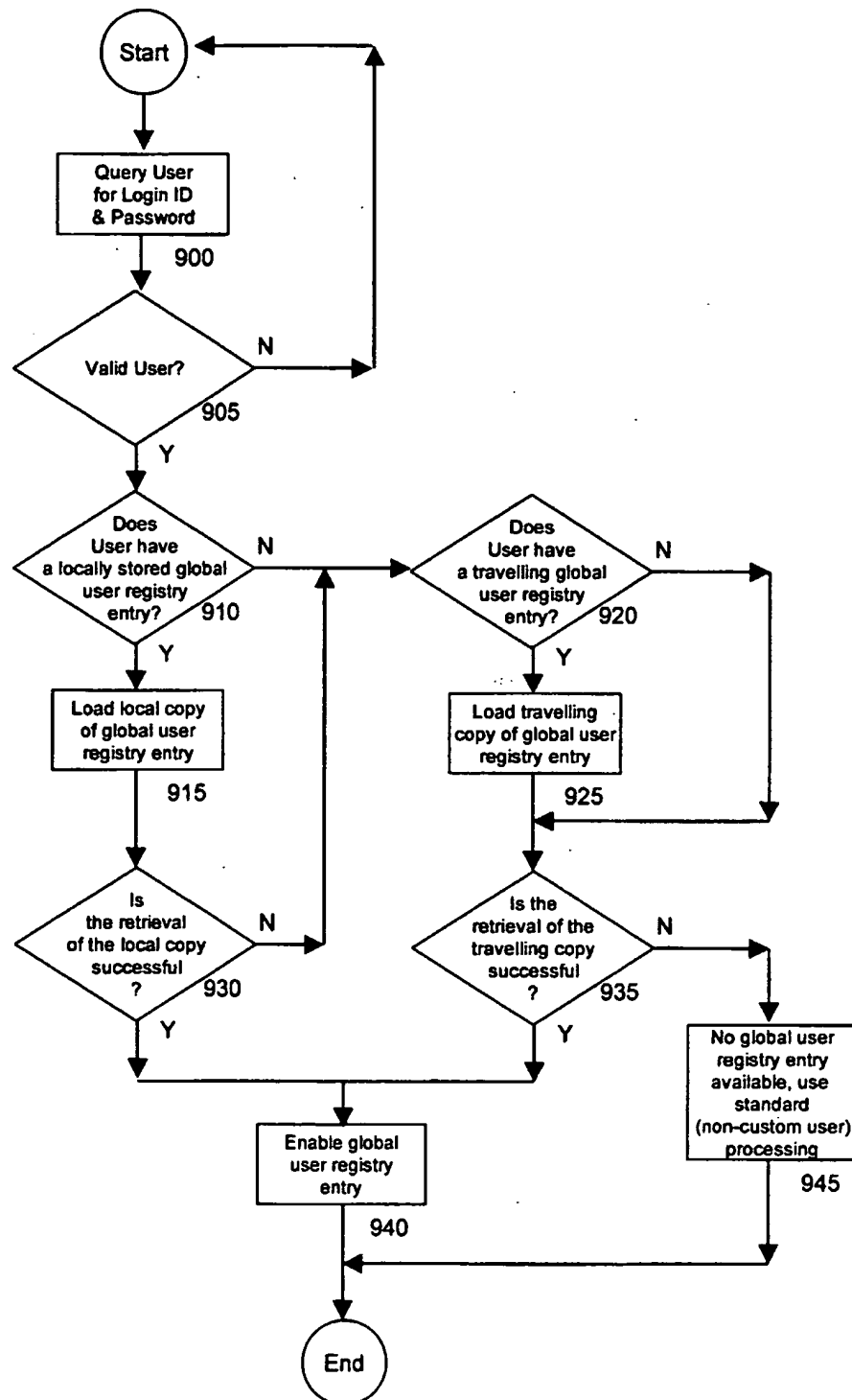


FIG. 10A

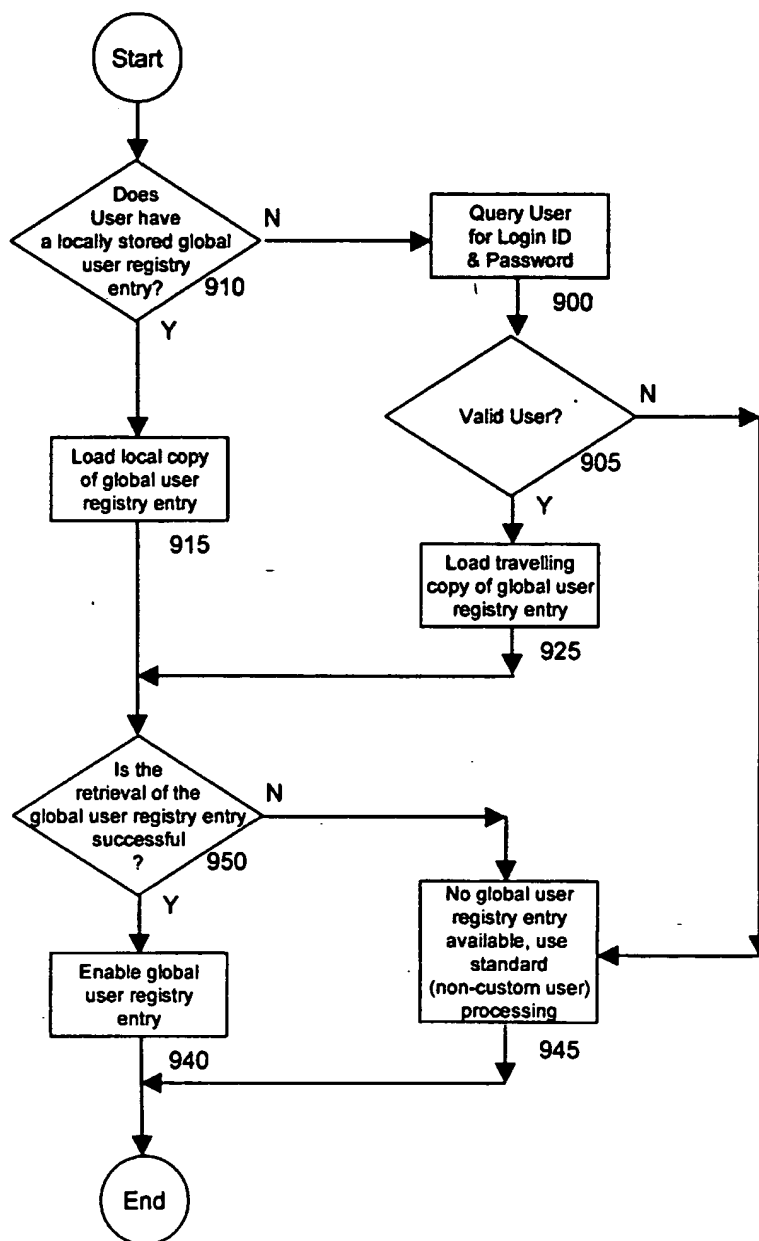


FIG. 10B

219



Log ID	User ID	Statement	Context	Date/Time
1686	42	www.pg.com		12/28/99 10:42am
1687	-1	"Would you like to hear about our products?"		12/28/99 10:42am
1688	42	"No."		12/28/99 10:43am
1689	42	"I would like to see movies."	Movies	12/29/99 03:42pm
1690	-1	"Would you like to see a theater listing?"		12/29/99 03:42pm
1691	42	"Yes."		12/29/99 03:42pm
1692	42	"How do I format this paragraph?"	Word processing	01/28/99 10:42am
1112A-x				
:				
1832	42	"Do you want to hear the news?"	News	01/28/00 10:11am

FIG. 11

## INTERNATIONAL SEARCH REPORT

In national Application No

PCT/US 01/11138

## A. CLASSIFICATION OF SUBJECT MATTER

IPC 7 G10L15/22 G10L15/26

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC 7 G10L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, INSPEC

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 00 14728 A (WEBER DEAN C ;ONE VOICE TECHNOLOGIES INC (US)) 16 March 2000 (2000-03-16) the whole document ---	1-37
X	WO 00 14727 A (WEBER DEAN C ;ONE VOICE TECHNOLOGIES INC (US)) 16 March 2000 (2000-03-16) the whole document ---	1,2, 5-21, 24-37
A	US 5 761 631 A (NASUKAWA ) 2 June 1998 (1998-06-02)  abstract  --- -/--	3-5, 15-18, 22-24, 34-37



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

## \* Special categories of cited documents:

\*A\* document defining the general state of the art which is not considered to be of particular relevance

\*E\* earlier document but published on or after the international filing date

\*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

\*O\* document referring to an oral disclosure, use, exhibition or other means

\*P\* document published prior to the international filing date but later than the priority date claimed

\*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

\*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

\*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

\*8\* document member of the same patent family

Date of the actual completion of the international search

29 August 2001

Date of mailing of the international search report

06/09/2001

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax (+31-70) 340-3016

Authorized officer

Quélavoine, R

## INTERNATIONAL SEARCH REPORT

In International Application No

PCT/US 01/11138

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WYARD P J ET AL: "SPOKEN LANGUAGE SYSTEMS - BEYOND PROMPT AND RESPONSE" BT TECHNOLOGY JOURNAL, BT LABORATORIES, GB, vol. 14, no. 1, 1996, pages 187-207, XP000554648 ISSN: 1358-3948 abstract -----	1-8, 19-27
A	US 5 311 429 A (TOMINAGA ) 10 May 1994 (1994-05-10) abstract -----	15-18, 34-37

**INTERNATIONAL SEARCH REPORT**  
Information on patent family members

International Application No  
**PCT/US 01/11138**

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 0014728 A	16-03-2000	AU 5813099 A	27-03-2000
		AU 6243999 A	27-03-2000
		EP 1110206 A	27-06-2001
		EP 1110205 A	27-06-2001
		WO 0014727 A	16-03-2000
WO 0014727 A	16-03-2000	AU 5813099 A	27-03-2000
		AU 6243999 A	27-03-2000
		EP 1110206 A	27-06-2001
		EP 1110205 A	27-06-2001
		WO 0014728 A	16-03-2000
US 5761631 A	02-06-1998	JP 2855409 B	10-02-1999
		JP 8147299 A	07-06-1996
US 5311429 A	10-05-1994	JP 2301869 A	13-12-1990